

# Manual for ZX-Tape

## Interface V1.02

ZX-Tape is a Bluetooth interface for connecting ZX Spectrum with PC for loading and saving programs. If you use any Z80 compiler on your PC, the ZX-Tape interface can download your hex/bin file directly into spectrum's memory and execute the code automatically.



## Table of Contents

1. Introduction .....	3
1.1 Key features .....	3
2. Installation and configuration .....	4
2.1 Connecting Bluetooth device: .....	4
2.2 COM port verification for ZXTape application: .....	5
2.3 Installation of ZXTape application .....	7
3. ZXTape application .....	8
3.1.1 Loading and Saving programs .....	8
3.1.2 Tap internal names and file names .....	9
3.2.1 Menu: File .....	10
3.2.2 Menu: Settings .....	10
3.2.3 Menu: Tools .....	10
3.2.4 Menu: About .....	11
3.3 List of programs in TAP directory on ZX Spectrum .....	11
4. Joystick interfaces Sinclair 1, 2 and Kempston .....	12
4.1 Configuration .....	12
5. DEBUG assembler under Zilog Developer Studio .....	13
5.1 Example of dbg.TAP file for DEBUG mode .....	14
5.2 DEBUG mode configuration .....	15
6. Write your own spectrum programs with Bluetooth HC-05 .....	17
6.1 Serial RS232 I/O Registers address space .....	17
6.2 Simple program in BASIC for transmitting data on RS232 .....	17
6.3 Simple program in BASIC for receiving data on RS232 .....	18
6.4 Is that all? .....	18
7. Schematics .....	19
8. Known issues in ZXTape application .....	20

## 1. Introduction

Dear customers, please let me introduce you to the ZX-Tape interface device. It was developed during COVID-19 lockdown 2020 for my kids to keep them entertained by learning a basic programming language. Even though the ZX spectrum in present days is more associated with an older generation, there will always be something magical in the simplicity of the design and games written for this machine. When architecture of modern computers grows in complexity, the way to understand computers and how they work for young enthusiasts becomes objectively more complicated. To teach my kids, I had to step back and recover my old ZX Spectrum from the attic. This machine might be old, but in spite of its simple architecture it maintains the same main principle of modern computers. Thus it will always be a good educational tool which even kids can understand.

Why did I decide to design the ZX-Tape interface device? The weakest point of the original ZX-Spectrum in modern days is how the programs are loaded and saved onto the tape. Anyone who decides to spend time writing programs for ZX-Spectrum, expects fast and reliable Load / Save commands. Similar products already exist on the market, capable of saving and loading programs directly from SD Card, so why create a new one with Bluetooth?

As the purpose of this device is mainly educational, the programmers should be able to write and test programs written in assembler language. Direct connection to the PC allows this to happen. The PC can run a programming tool called Zilog Developer Studio for writing assembler code. The ZXTape application will automatically load the program into ZX Spectrum after successful compilation. The result can be seen almost immediately (see DBG mode in this manual for further details). This advantage can't be achieved with other interfaces on the market. In addition, with connection to the PC, you can play games from the internet straight away after download. There is no need to learn new commands as the Load / Save works exactly as you would use an old cassette recorder. Furthermore, Bluetooth serial communication hardware can also extend your own programs.

### 1.1 Key features

- The basic program of ZX Spectrum in ROM operates exactly the same way as per original. No additional or special commands are introduced.
- Loading and saving programs in TAP file format over the Bluetooth to your windows PC running a ZXTape application. Supported OS versions: Windows XP<sup>1</sup>, Windows 7, 8 and 10.
- 2x Joystick interfaces configured as: Sinclair1 and Sinclair2 or Kempston and Sinclair2.
- Use of standard removable HC-05 Bluetooth module in RS232 mode with baud rate of 115200.
- Description of registers, so the Bluetooth communication can be used in custom programs.
- DBG mode: Write assembler under Zilog Developer Studio with a fast upload into ZX-Spectrum (Like Matthew Smith during the development of Manic Miner).
- Supported models: ZX Spectrum 16kB/48kB

**[1] Note:** If the Bluetooth stack installed on Windows XP is from another company than Microsoft, the application cannot detect the presence of ZXTape Bluetooth device. If that is the case, you have to close and re-open ZXTape application every time you disconnect ZX spectrum from the power. Remember, ZX-Tape interface includes RESET button, so normally there is no need to disconnect ZX spectrum from power in order to reset it.

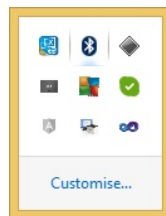
## 2. Installation and configuration

Before you can use the ZX-Tape interface, it has to be connected via Bluetooth with your windows PC. The Bluetooth device is a separate module HC-05 which can be detached from the ZX-Tape interface. It provides serial port connection between ZX-Spectrum and your PC. The HC-05 module comes pre-configured to: 115200 baud rate, 8 data bits, 1 stop bit and no parity. Device name is set to: ZXTape. These parameters cannot be changed in order to ensure the device operates with ZXTape application correctly.

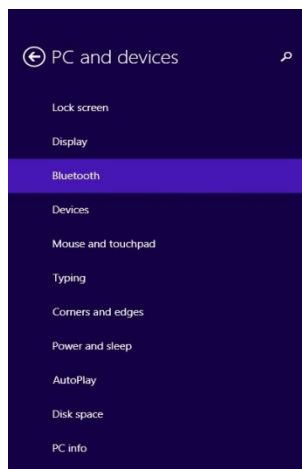
Please, follow the instructions below to connect the ZX-Tape interface with your PC. The pictures shown are taken from Windows 8 and can differ from other versions of windows operating systems. There are several ways to set up connection with a Bluetooth device. Here is one of them:

### 2.1 Connecting Bluetooth device:

- Insert ZX-Tape interface hardware into you ZX Spectrum and power it on.
- On the bottom bar of the windows PC, click the small arrow on the right hand side to reveal small icons:

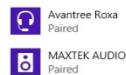


- Click the Bluetooth icon and select Show Bluetooth devices. After the ZX Spectrum is powered on, an unknown device icon will appear on the screen, this will later update to ZXTape:



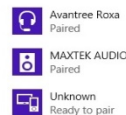
#### Manage Bluetooth devices

Your PC is searching for and can be discovered by Bluetooth devices.



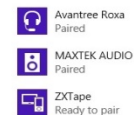
#### Manage Bluetooth devices

Your PC is searching for and can be discovered by Bluetooth devices.

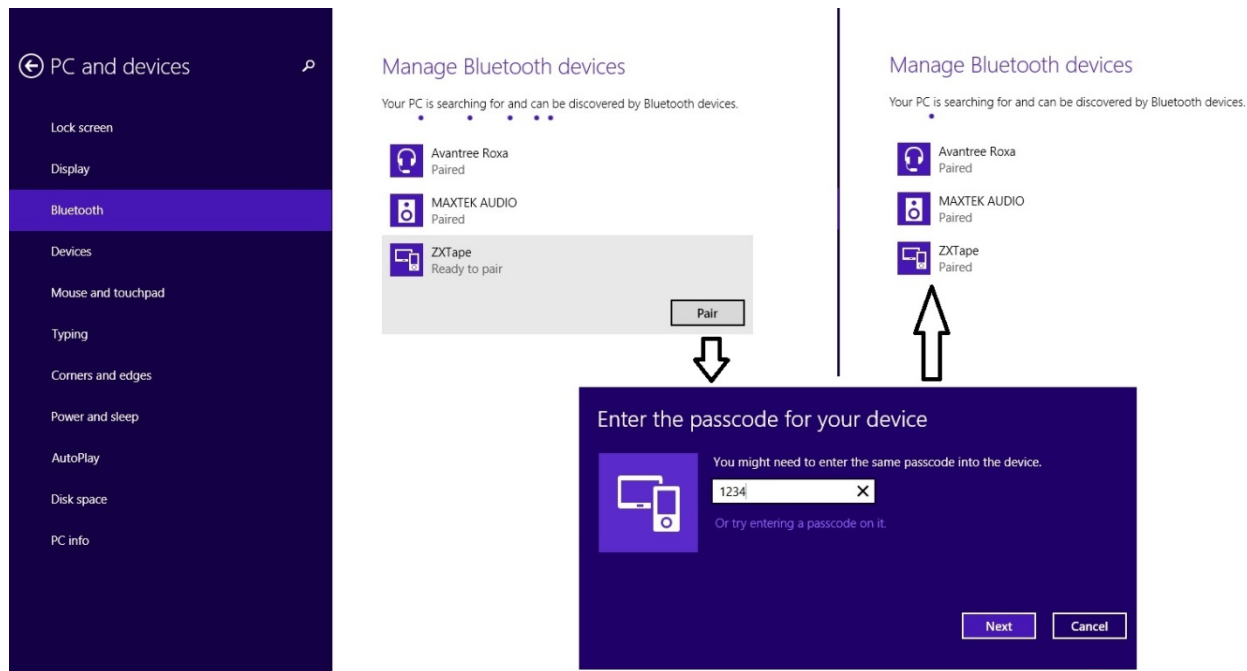


#### Manage Bluetooth devices

Your PC is searching for and can be discovered by Bluetooth devices.



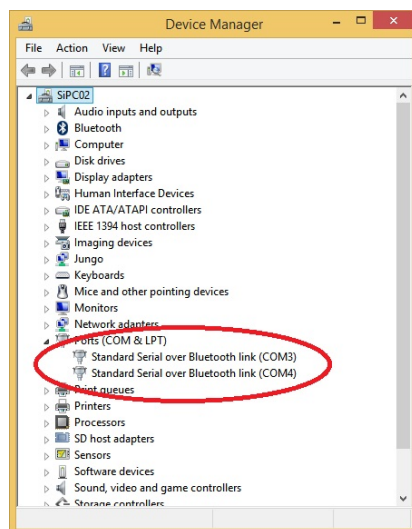
- d) Click the Pair button on ZXTape device and enter code: 1234 when prompted. If the pairing process was successful, the status will change to Paired:



Once the Bluetooth device is paired, there is no need to pair it again. When the ZX-Tape interface is powered and within good radio reception, then the ZXTape icon will appear in the Bluetooth device window with a status: Connected. The distance for good reception varies between 5 and 10 meters. It also depends on obstacles in-between the peers, for example a wall can reduce Bluetooth reception.

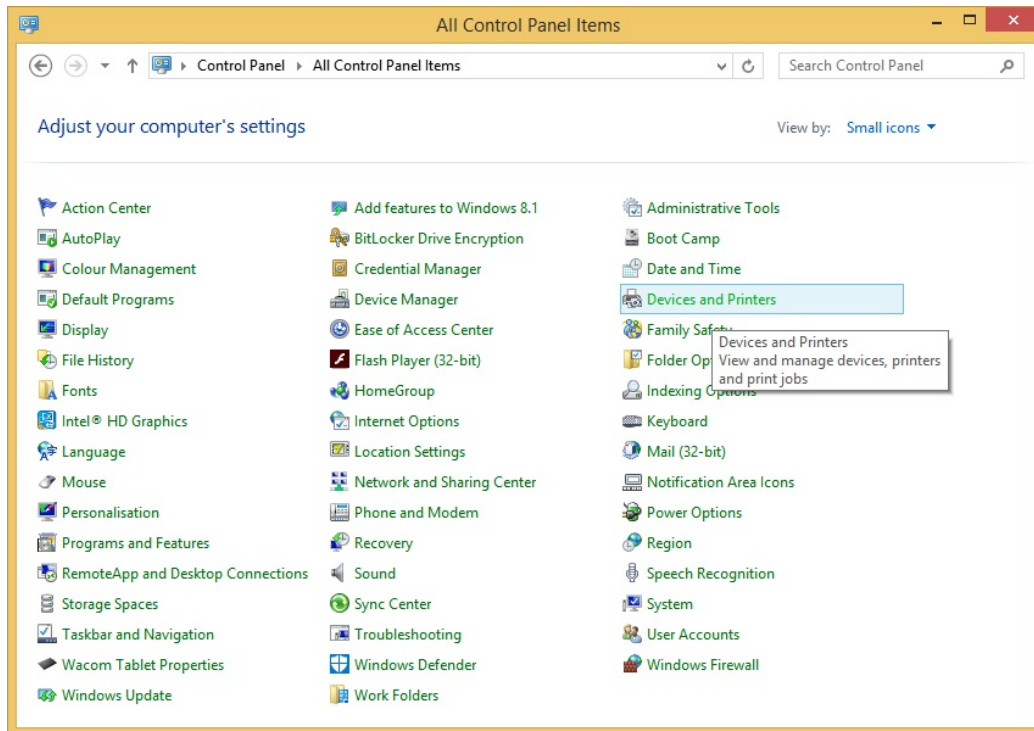
## 2.2 COM port verification for ZXTape application:

When the Bluetooth is paired, you should see in the Device Manager under Ports (COM & LPT) two new com ports:



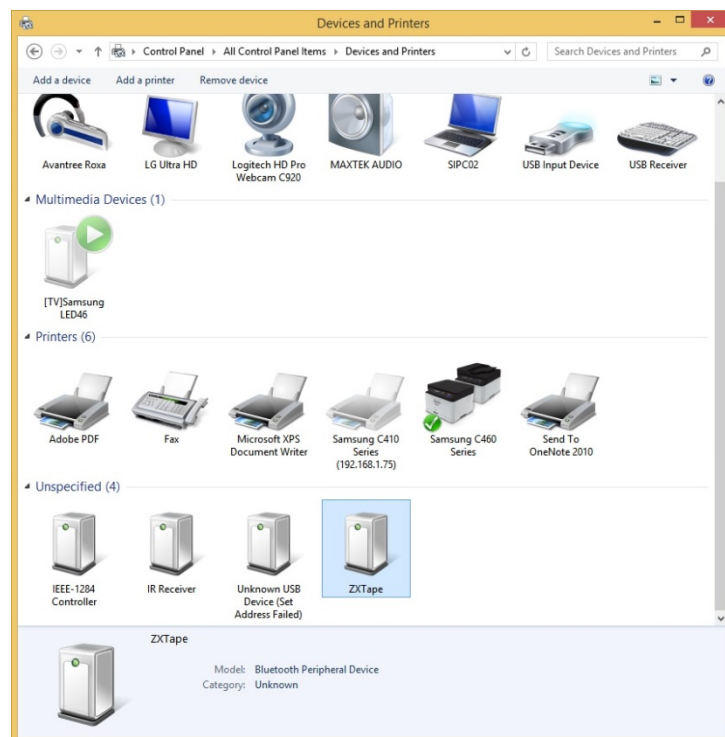
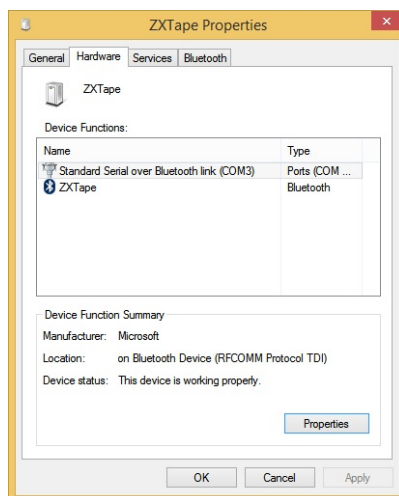
Although two com ports appear, only one of them is used as a serial line. Our example here shows COM3 and COM4.

You could try out both serial ports in the ZXTape application to see which one will work. Alternatively, continue reading this section to explicitly identify the correct one. Open the Control Panel using **+X** and then click on Devices and Printers:



Then scroll down to unspecified devices and double click on ZXTape device:

When the ZXTape properties window is shown, click on Hardware tab to find out the COM port number for the Bluetooth. In our example it is COM3:



## 2.3 Installation of ZXTape application

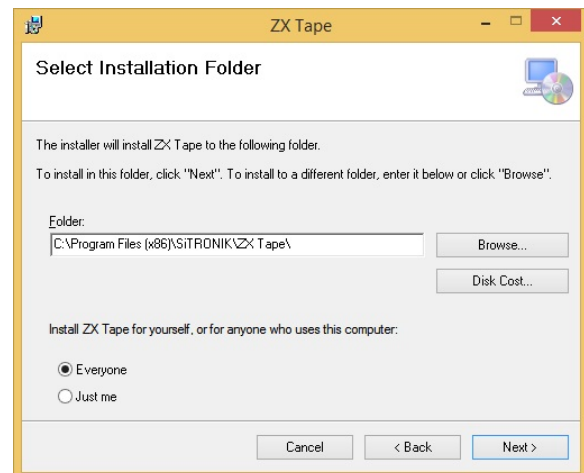
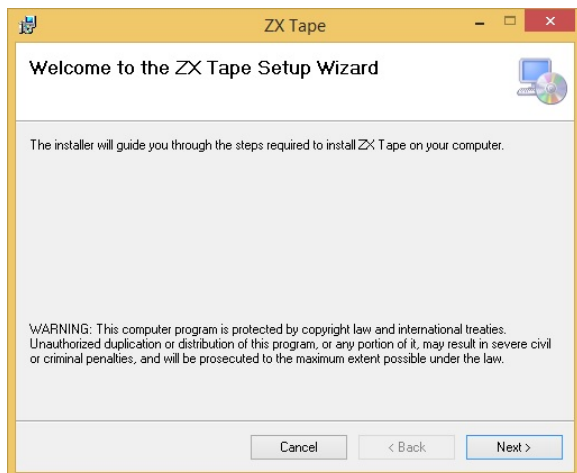
Download compressed installation files from the following web site:

[www.sitronik.co.uk/ZXTape/software.zip](http://www.sitronik.co.uk/ZXTape/software.zip) and extract software.zip file. After the extraction you should see 2 files:

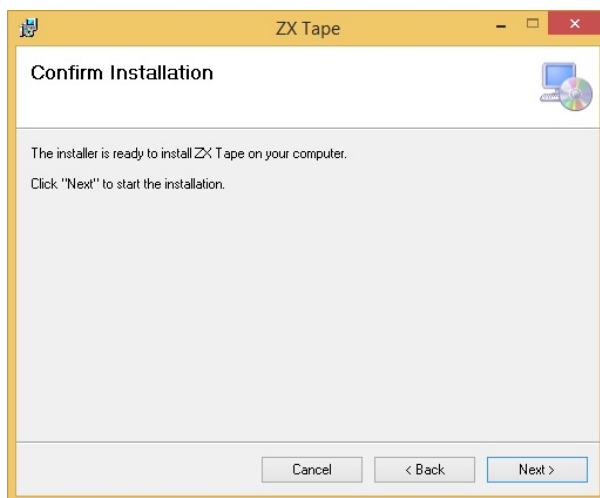
Setup ZX Tape.msi  
setup.exe

Follow next steps to complete installation of ZXTape application.

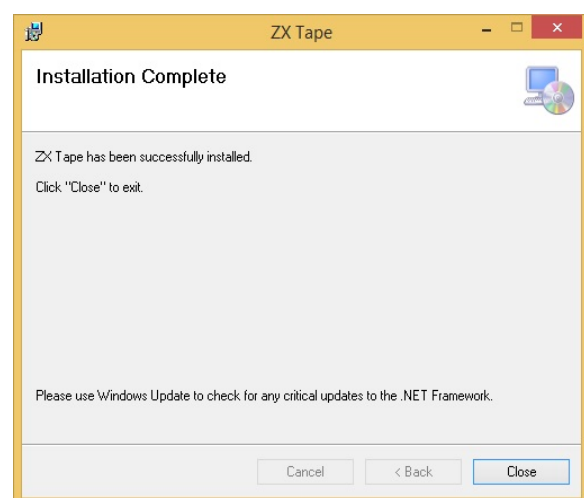
1. Double click on setup.exe file and then click NEXT > button when Setup Wizard window appears.
2. Choose a folder where you wish to install the application and which users can access it. Then Click NEXT > button.



3. Click NEXT > button to confirm installation.

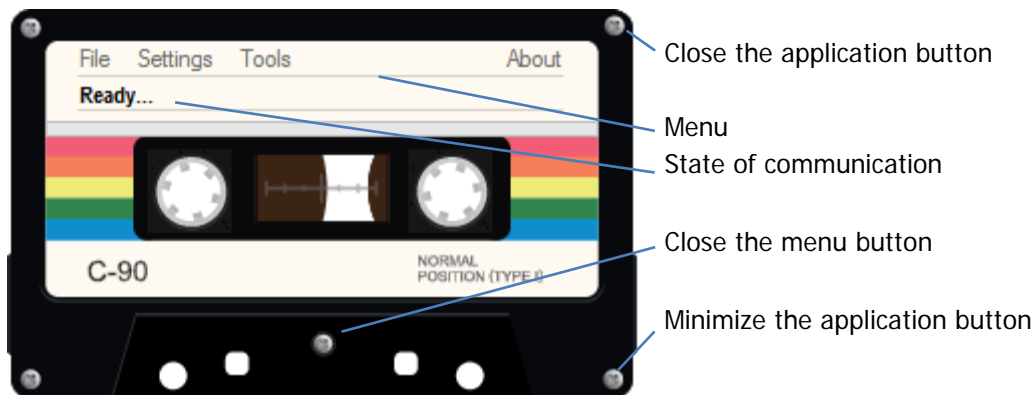


4. Click Close to exit. The ZXTape application is now installed.





### 3. ZXTape application

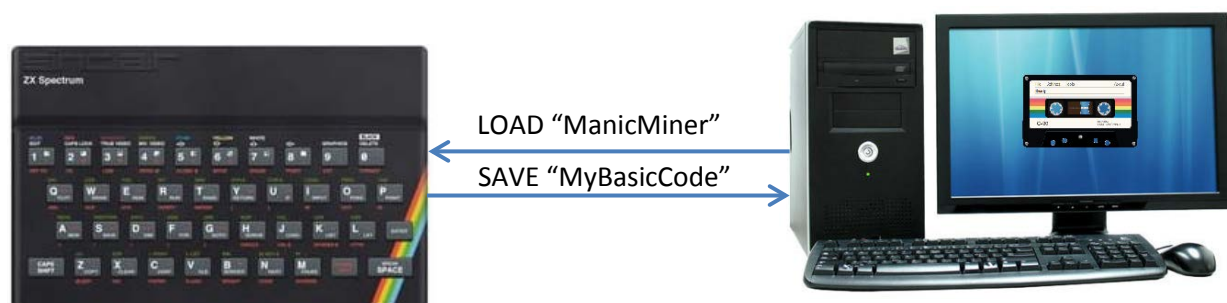


The ZXTape application is a windows software in the shape of cassette tape. Its main purpose is to simulate a tape recorder. It can be also used for loading binary or hex files produced by assembler or C compiler on your PC directly into memory of ZX spectrum for further debugging purposes. The hard drive of your computer will serve as storage for your programs. ZXTape application supports files in TAP format only. This format is the closest to that used to store data on the real tape. Most of the programs and games you can download from internet websites, like worldofspectrum.org are in this format already. In case of TZX format (a more general format which includes other machines from 8bit world) you can use converter from the menu **Tools** for conversion to TAP file.

The most important feature of this product is that no new commands are introduced into BASIC/Monitor firmware. The aim is to give you something that you are used to. The LOAD and SAVE commands will work exactly the same old way. Their routines were overwritten in ROM to work with standard serial port RS232 over Bluetooth and few lines of code have been added to support downloading assembler directly into memory of ZX Spectrum. This feature can help you easily improve your assembler skills too. I hope that this simple hardware will cover all aspects of educational tool.

#### 3.1.1 Loading and Saving programs

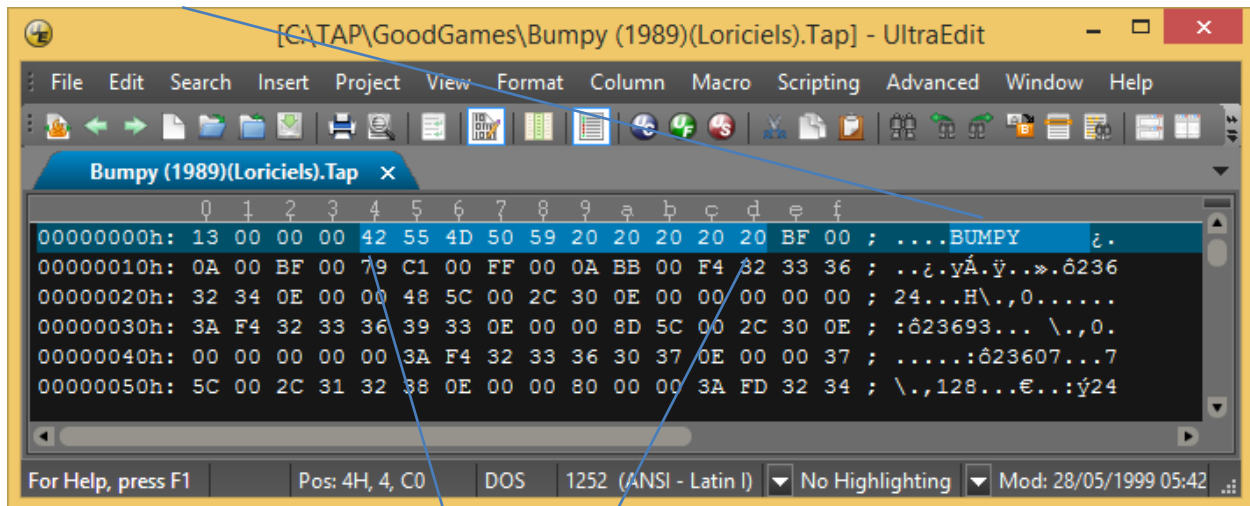
After correct installation of the ZX-Tape interface you can use LOAD and SAVE commands straight away. All spectrum programs will be loaded and saved into the TAP directory you choose in the menu **Settings**. See section 3.2.2.





### 3.1.2 Tap internal names and file names

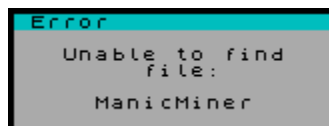
ZX-Tape is fully compatible with your old ZX Spectrum, so when you use the command LOAD "ProgramName", then the parameter "ProgramName" must correspond to the name inside of the data file, not to the name of the file. Some programs downloaded from the internet might not have the file name identical to internal name of the TAP file. For example you can download the game Bumpy (1989)(Loriciels).TAP The filename is called Bumpy (1989)(Loriciels), but its internal name inside of the TAP file is "BUMPY". See picture below:



Each TAP file has the internal name of the program hidden here and you use it with command LOAD or it is created with command SAVE. Names are **case sensitive** with maximum length of 10 characters. Inside of the TAP file are located at position 4 to 13. Shorter names are filled with space character 20h up to position 13.

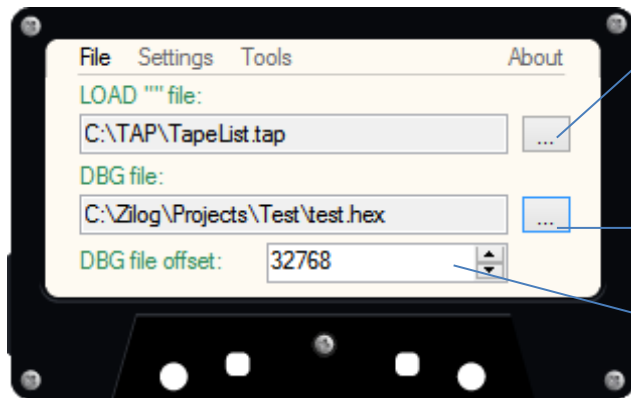
You don't have to worry that command LOAD "BUMPY" will not work as the ZXTape application searches for internal names. But this inconsistency between file names and internal names can become pretty confusing. If you want to keep it right, use utility in menu **Tools** for renaming files to their internal names. See section 3.2.3. Remember that SAVE command will always create a filename identical to the internal name.

If the program with the name you enter doesn't exist in your TAP directory, you will get an error message window on your ZX Spectrum screen:



### 3.2.1 Menu: File

In the old days, when you used the LOAD command without entering the program name: LOAD "", you could load the first program which appeared on your cassette tape. The ZXTape application is always expecting the name of the program to be entered or you can set a default TAP file in the File menu to be loaded if no program name was specified. This can spare lots of typing when loading your favorite game.



Select a default TAP file which will be loaded when command: LOAD "" without file name is executed.

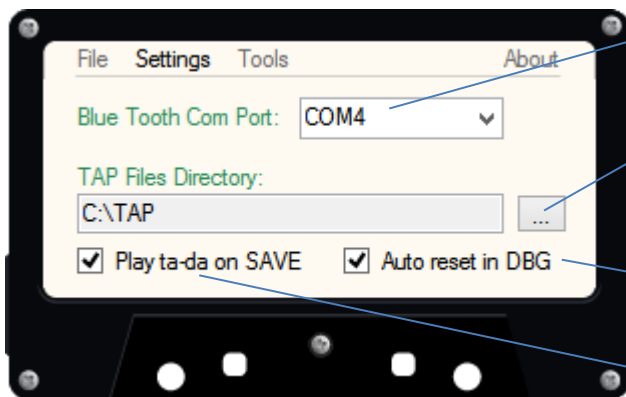
Enter name of the HEX or BIN file produced by assembler after compilation. See DBG mode in section 5.

Enter offset in bytes where the HEX or BIN file data will be read from.

When working with assembler in DBG mode, you need to specify the name of the output file after compilation. It can be either HEX or BIN file. The ZXTape application continuously checks the time stamp of this file and as soon as it is changed, it forces a reset of ZX Spectrum in order to download this file.

### 3.2.2 Menu: Settings

In this menu, you must set the key parameters like com port and TAP directory in order to get the application to work. Once it is set correctly, the tape wheels on the main window will start turning and status shows that ZXTape is **Ready...**



Selects COM port number for ZX-Tape Bluetooth. See section 2.2 to select correct com port.

Selects folder where you store all TAP files. This directory is used when you enter Load or Save command with a file name in argument.

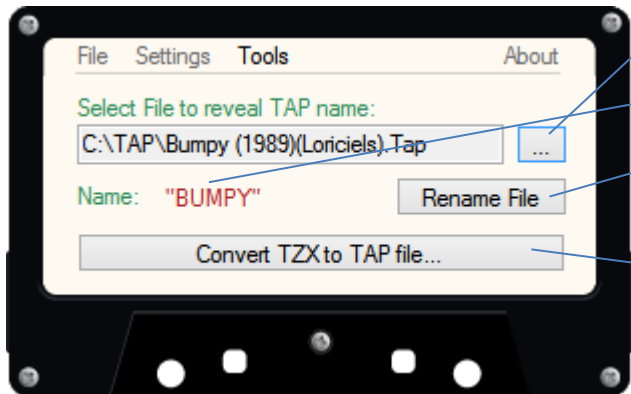
This option will automatically reset ZX Spectrum and start loading HEX or BIN file after it has been updated. This only works in DBG mode.

Plays ta-da song after successful execution of SAVE command.

### 3.2.3 Menu: Tools

ZXTape application works only with TAP files. The name of the TAP file is case sensitive and must be identical to its internal name of the program. This is the name you enter in SAVE "name" or LOAD "name" command between quotes. It is maximum 10 characters long and can be found after the 4<sup>th</sup> byte from the beginning of the file when you open it in binary viewer. For more detailed information see: [https://faqwiki.zxnet.co.uk/wiki/TAP\\_format](https://faqwiki.zxnet.co.uk/wiki/TAP_format).

Other formats are not supported. That's why the ZXTape application is offering a few tools which can help you with renaming TAP file names to their internal names as well as convert them from TZX format to TAP format. See picture with a description below:



Selects file for revealing its original TAP name.

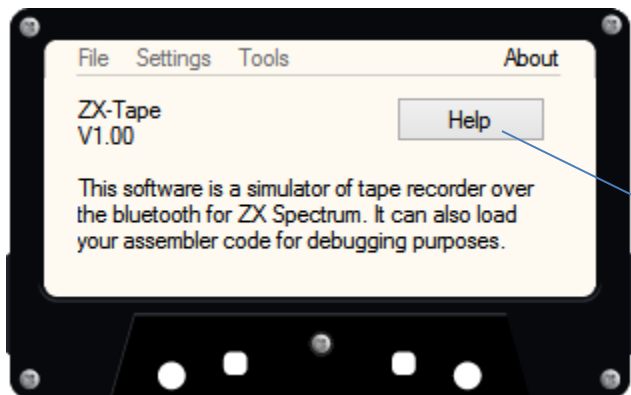
Revealed name of the original TAP file.

Rename the TAP file to its original TAP name.

Converts TZX files into a TAP file format, so it can be used in the application. Supports only ZX-Spectrum up to 48kB.

**Note:** After conversion, you might still use the above utility to rename TAP file to its original name.

### 3.2.4 Menu: About



This window provides information about the software version and short description of use.

Help button for opening this manual as a pdf.

### 3.3 List of programs in TAP directory on ZX Spectrum

To view all programs in your TAP directory on ZX Spectrum, you can use the program **TapeList.TAP**. When you run this program, ZXTape application will generate data with all internal TAP file names inside your TAP directory in alphabetical order so it can be displayed on the ZX Spectrum. This can be useful if you can't remember the name of the program you wish to run. Just use the cursor keys to navigate to the program you want to run and press ENTER. After the installation, option LOAD"" file, in menu File, has the TapeList.TAP selected already. To bring the list of files on the screen, just type LOAD "".

<<-0	Page: 1/3	P->>
Arena1	camelot	Galax1
arrows	CANNON	HOPPER
AsteroidB	CBOX	horace
AUTOCRAS	Chopper	HYBRID
BACKGAMMON	CHUCKIE	HYSTERIA
DREAM	CLOCKMAN	ManicMiner
BANGERS	COLONY*	p31
Batman	Cosmos	PAC-MAN
BBunny4	Golf	POSTMAN
Universa	DANGEROUS	PERISCOPE
BLOCKMAN	DARTS	PHANTOMAS1
BLOODGUTS	DEFLEKTOR	PIPELINE
Jack	DETORN	Q-BERTUS
BOMBARDER	DizzyDice	raiders
bomber	dotty	Sidewize
Bomb Jack	E L I T E	SKIING
BONES	almirante	Spinvader2
Booty1	ELIMINATOR	TETRIS 48
BORER DEEP	ESKIMO	TUNNEL
Bridge	FARMER	PAC-MAN
BU-RIL	F & F	Loader
BUMPY	FOOTBALL	SECRET
CABMAN	FREEDOM	SKULL

## 4. Joystick interfaces Sinclair 1, 2 and Kempston

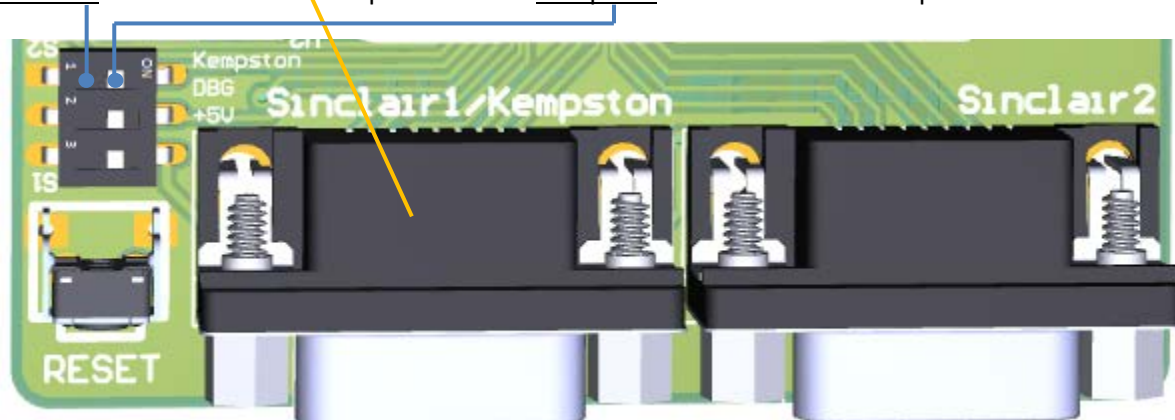
For game players the ZX-Tape interface is also supporting two Sinclair joystick inputs where the first one can be configured as Kempston.

Interface	Port Address *  [Hex]	Direction	Data								Notes	
			MSb		Bits							LSb
			7	6	5	4	3	2	1	0		
Sinclair 1 CONN3	EF FE	IN	Hi	Hi	Hi	Key – 7 on Pin4	Key – 6 on Pin3	Key – 8 on Pin2	Key – 0 on Pin6	Key – 9 on Pin1	When one of these port addresses is accessed during keyboard scanning routine on ZX-Spectrum, the interface uses open collector output on the data bus to fetch information about the joystick state. It simulates the pushing of specific keys if the appropriate pin is set to logic Lo or the pin is grounded.	
Sinclair 2 CONN2	F7 FE	IN	Hi	Hi	Hi	Key – 2 on Pin4	Key – 1 on Pin3	Key – 3 on Pin2	Key – 5 on Pin6	Key – 4 on Pin 1		
Kempston CONN3	XX1F	IN	Hi-Z	Hi-Z	BUTTON2 on PIN9	BUTTON1 on PIN6	UP on Pin1	DOWN on Pin2	LEFT on Pin3	RIGHT on Pin4	When this port address is accessed during IO operation, the interface transfers 8 bits of data as a totem pole output on the data bus. Bits 6 and 7 are always set to logic Lo or ‘0’. The logic level Hi or ‘1’ is read if the appropriate pin is set to logic Lo or if the pin is grounded. Otherwise if the input pin is left open, the data bit reads logic Lo or ‘0’.	

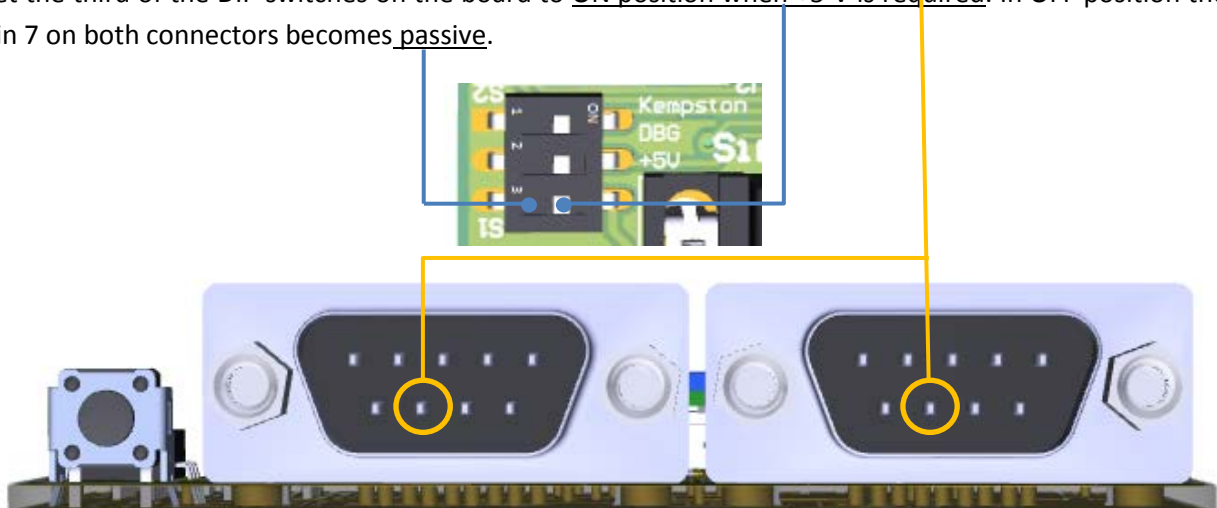
\* X-don't care bits

### 4.1 Configuration

Joystick connector CONN3 can be configured by first of the DIP switches on the board to operate as: Sinclair1 when switch is in OFF position or as Kempston when switch is in ON position.



To be compatible with joysticks or joypads from different manufacturers, pin 7 on both connectors CONN2 and CONN3 can be configured to supply +5 V from the ZX-Spectrum to the joystick's circuitry. For example joysticks with an automatic fire button need active electronic parts inside to make it work. Set the third of the DIP switches on the board to ON position when +5 V is required. In OFF position the pin 7 on both connectors becomes passive.



## 5. DEBUG assembler under Zilog Developer Studio

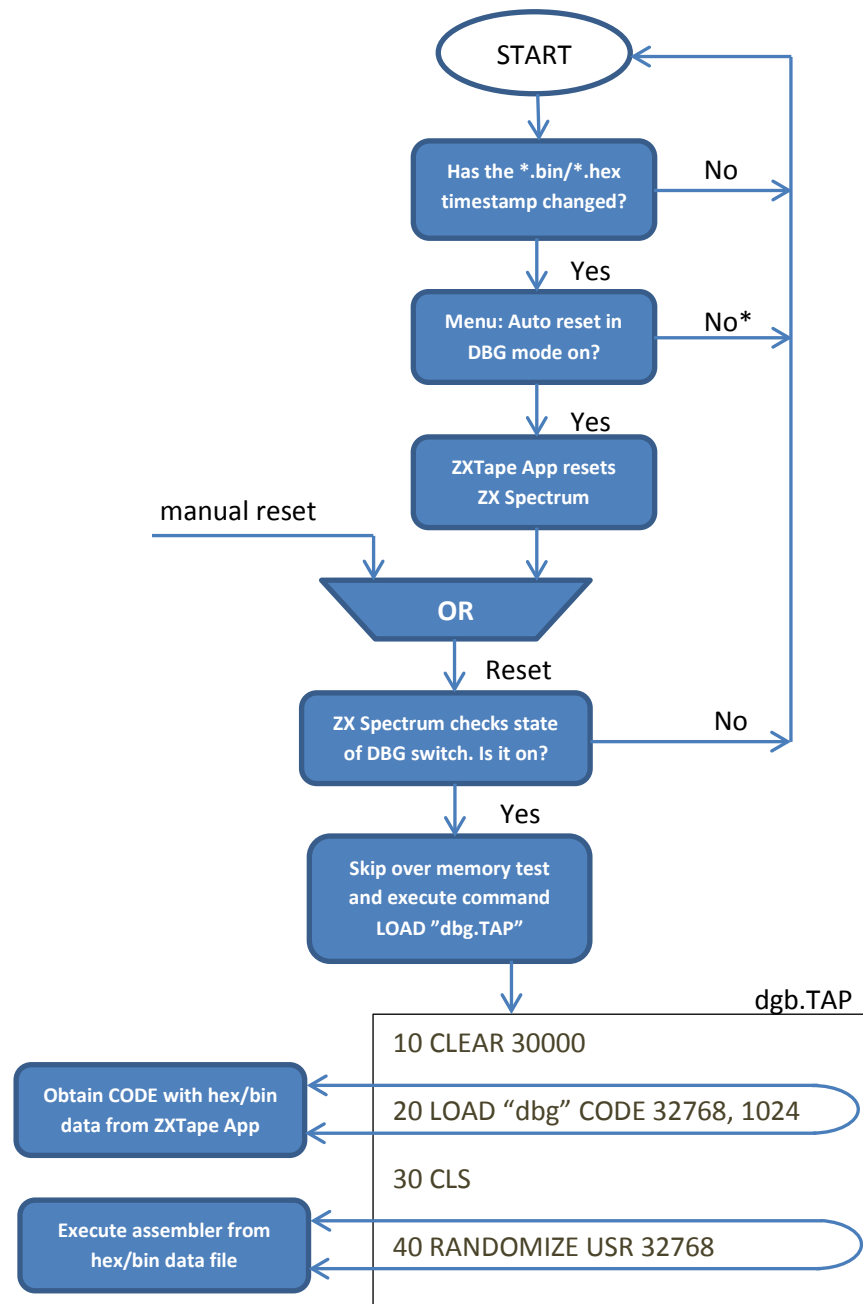
When you start mastering BASIC programming you will soon realize that some routines are not as fast as you would expect. For example BASIC is not good at binary operations. When you look closely at the program in chapter 6.3, reading of bit 0 from the register takes 2 lines of BASIC code and it also looks a bit complicated. The same task when written in assembler requires only a few instructions. Also if you imagine that each BASIC command line consists of hundreds of assembler instructions, you will realize that the cost of easy interpretation of BASIC language reflects in its speed.

To speed up their programs many programmers are combining BASIC with assembler routines or using pure assembler code. One of them is Matthew Smith who developed the famous game Manic Miner completely in assembler language. He reconnected his ZX spectrum via parallel cable with a TRS-80 PC from where he sent the assembler program directly to the memory of ZX spectrum for further debugging.

The ZX-Tape interface in DBG mode is capable of doing the same thing. It sends assembler code compiled on your PC into memory of ZX spectrum and starts it there.

Let's assume you are using Zilog Developer Studio I version 3.68 downloaded from zilog.com as your assembler compiler, and ZX-Tape interface configured for DBG mode. Every time you successfully compile your assembler in the developer studio, you get a new binary (hex) file as an output. If you set the path of this file in ZXTape application, then the application will start checking the timestamp of the file. As soon as it changes, the application will reset ZX Spectrum and start loading new binary (hex) file

with your assembler program into memory of ZX Spectrum, continuing with its execution. This feature will help you learn to develop and debug your own assembler routines. See the following diagram:



\* When the Auto reset in DBG mode is off, you must manually reset the ZX Spectrum to load assembler. This can be useful in case that your assembler is causing a reset of the machine and thus reloads itself in an infinite cycle.

## 5.1 Example of dbg.TAP file for DEBUG mode

After the installation of ZXTape application, this file is located in "C:\TAP\dbg.TAP". It is loaded into ZX Spectrum after manual or automatic reset of the machine in DBG mode. Example of the code:



```
10 CLEAR 30000
20 LOAD "dbg" CODE 32768, 1024
30 CLS
40 RANDOMIZE USR 32768
```

If you want to combine assembler with BASIC, first you need to make sure that your assembler code will not interact with BASIC memory space. Using CLEAR 30000 on the first line sets the RAMTOP variable to memory address 30000. This RAMTOP variable is limiting working space for basic up to address 30000. After this command all memory space above this address will become available for use in your assembler program without interfering with BASIC and vice versa.

The second line loads your assembler starting at address 32768 in Spectrum memory with a length of 1024 bytes. If your assembler program is longer than 1024 or you wish to start your assembler code at a different address, then you must modify your dbg.TAP file.

The next line clears the screen and then the last line executes your assembler at address 32768.

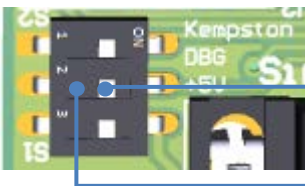
In case you need to modify any of the parameters in dbg.TAP, use save command with LINE argument which will automatically start BASIC program dbg at line 10 after loading:

SAVE "dbg" LINE 10

Make sure your modified dbg.TAP file is in the TAP directory, and configuration of ZXTape application is correct before you start writing and debugging your own assembler.

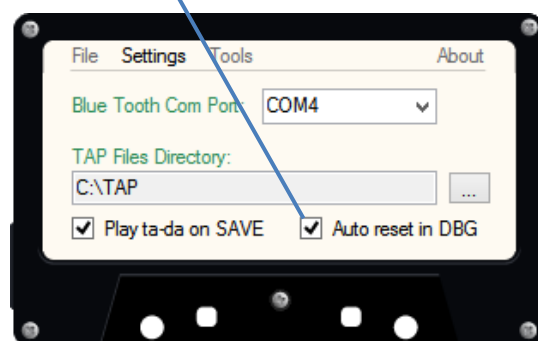
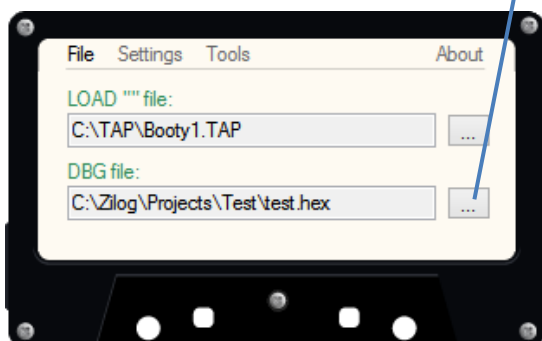
## 5.2 DEBUG mode configuration

For correct configuration of the ZX-Tape interface for debugging you assembler code follow these steps:

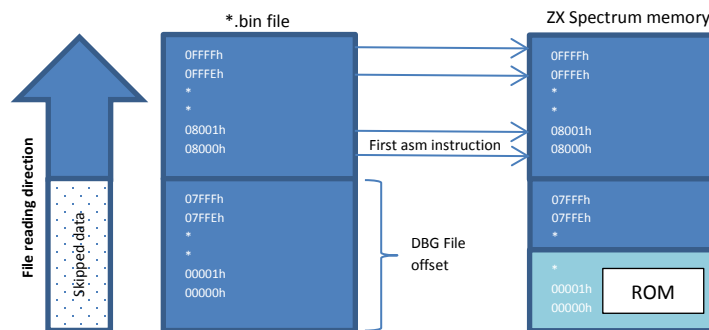



1. Set the second of the DIP switches on the board to ON position to put ZX-Tape interface into DEBUG mode. For normal operation push it to OFF position.

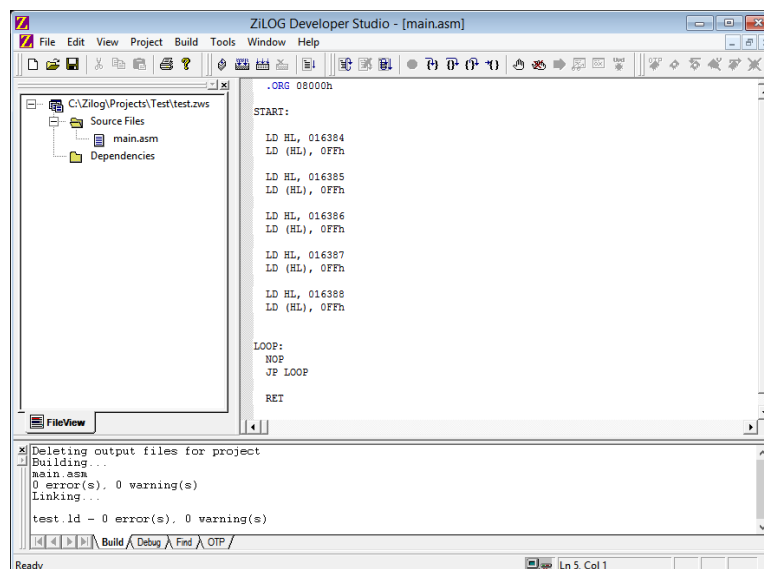
2. In ZXTape application File menu, select hex or bin file produced by assembler after compilation. Then in the Setting menu select if you wish for the app to automatically reset and load your assembler.



3. Open Zilog Developer Studio and create a testing assembler program called main.asm. See picture below. The program will write value 255 into the first line of the video memory from address 16384 to 16388 and then it enters an infinite loop. It is important to write line: `.ORG 08000h` at the beginning of the assembler file. This will force the compiler to put the first instruction of your code at address 08000h in the binary file or more precisely in the ZX Spectrum memory. Remember that the ZXTape application is reading binary file from the offset you can set in File menu. After compilation, the Zilog Developer Studio will fill address space between 00000h and 07FFFh with zeros in the file. After that it continues with program data. This is because the `.ORG` directive tells the compiler to place your code from address 08000h. This address must match first parameter of basic LOAD command in dbg.TAP file on line 20. See previous section 5.1. Furthermore, we also need to set DBG file offset in the File menu to 32768 (08000h). This will force ZXTape application to start reading the binary file from address (with an offset in the file) 32768 (08000h), where the program's data is located. Some other compilers like z88dk are creating binary files without filling zeros at the start of the file, but still expect that the program data will start at address 32768 (08000h). In such case we must set the DBG file offset in the menu to 0. Picture below shows the situation in our example:



When it is all correctly set, hit the button  to compile your assembler. If the Auto reset in DBG is selected, ZXTape application will automatically load your assembler into ZX Spectrum. Now, as a result, if you see a black horizontal line on the TV screen, your program has worked!



## 6. Write your own spectrum programs with Bluetooth HC-05

If your programming skills are improved and you are looking for something more advanced to experiment with, then you can look at the ZX-Tape interface. It provides simplified functionality of UART - universal asynchronous receiver-transmitter serial interface similar to TL16C550 circuit. Due to the limited number of macrocells in ATF1508 CPLD, this simplification was made by omission of handshaking signals which are not required for Bluetooth module HC-05 and by eliminating FIFO buffer. For more details about the register map, see the next chapter or check the schematics diagram at the end of this manual.

### 6.1 Serial RS232 I/O Registers address space

Register Name	Port Address *  [Dec]	Direction	Data								Notes
			MSb		Bits				LSb		
			7	6	5	4	3	2	1	0	
Receiver & Transmitter Register  RTR	37	IN/OUT	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0	RTR - register is 8 bit wide. After write operation, data are transferred to the shift register and transmitted on TX pin at a baud rate of 115200. Check <b>TE</b> flag to ensure previous data was already transmitted before writing to this register. When the serial data is successfully received indicated by <b>DR</b> flag, use read operation on this register to fetch received data byte.
Line Status Register  LSR	55	IN	DBGsel	TE	THRE	0	FRAME_ERR	0	OVERRUN_ERR	DR	<b>DR</b> – data ready flag is set to 1 when RTR register received byte. Cleared after reading RTR register. <b>OVERRUN_ERR</b> – set when RTR register still contains previous data which hasn't been read yet. <b>FRAME_ERR</b> – is set when invalid stop bit was received and cleared after LSR register was read. <b>THRE</b> – transmitter holding register empty flag and <b>TE</b> – transmitter empty flag are identical due to the limited resources. When set, the RTR register can accept new data to be transmitted, otherwise it is cleared. <b>DBGsel</b> flag is not part of the serial line concept. It monitors the position of the DBG switch on the board.

### 6.2 Simple program in BASIC for transmitting data on RS232

This program is transmitting ASCII character of the key pressed to the serial line.

```

10 CLS
20 LET k$ = INKEY$
30 IF k$="" THEN GO TO 20
40 PRINT AT 0, 0; k$
50 OUT 37, CODE k$
60 IF INKEY$<>"" THEN GO TO 60
70 GOTO 20

```

**Note:** In assembler, a similar program would run much faster, so checking TE bit would be essential.



### 6.3 Simple program in BASIC for receiving data on RS232

This program is receiving ASCII characters from the serial line and displaying them on the screen.

```
10 CLS
20 LET s = IN 55
30 LET half = INT(s/2)
40 IF (s-2*half) = 0 THEN GO TO 20
50 PRINT AT 0, 0; CHR$( IN 37)
60 GO TO 20.
```

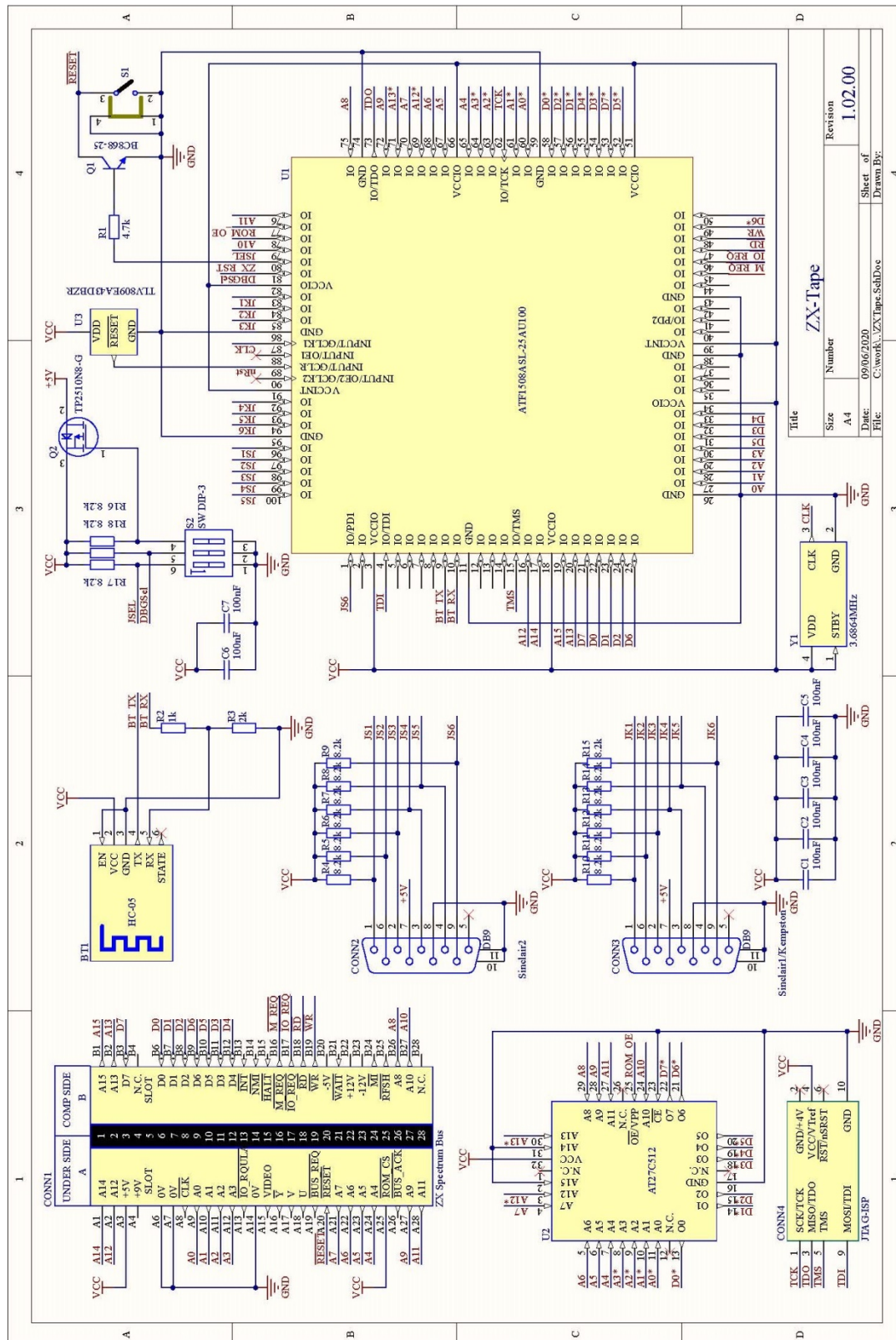
**Note1:** We use a little trick here to identify Bit 0 – Data Ready of LSR register. Least significant bit says whether the number is odd or even. Line 30 divides the value of status byte by 2 and assigns to variable half only the integer part of the number after division. On line 40 we check if there is a remainder. If so, Bit 0 was set and we continue in the program.

**Note2:** Make sure that your serial terminal program on PC is transmitting characters immediately when the key is pressed.

### 6.4 Is that all?

No, you can go even further. Through special configuration the Bluetooth HC-05 module offers more options than its original use. The content of the ZX-Tape interface package includes a manual for HC-05. You can find information there about one of the interesting configurations of this module called Peer to Peer. If you have a friend with another ZX-Spectrum, you can use your Bluetooth modules in this configuration, so you will be able to exchange data between each other. Does it ring a bell? Well, imagine you can create games for 2 players like a ship battle!

## 7. Schematics



## 8. Known issues in ZXTape application

Software version 1.00:

- When the power to ZX-Spectrum is interrupted the Bluetooth communication can be lost for a while. It can take 20 seconds before the ZXTape application recognizes this situation and displays: No Connection... When the power is back on, it will reconnect in 3 seconds. Please wait until the ZXTape application will establish connection again showing state: **Ready...**

Document revision	Date	Changes
Rev 1	5/12/2020	First release of this document.
Rev 2	29/12/2020	Information added about supported Windows OS versions. Small text corrections.
Rev 3	9/1/2021	Added information about DBG file offset in section 3.2.1 and corrected example in section 5.2